Southampton Solent University

Assessment Brief

Assessment Details

Unit Title:	Software Design and Development
Unit Code:	COM714
Unit Leader:	Prins Butt
Level:	7
Assessment Title:	Solent Excursions
Assessment Number:	AE1 – 4th Sitting
Assessment Type:	Software Artefact and Demonstration
Restrictions on Time/Length:	Software Artefact + 15 minutes demonstration
Individual/Group:	Individual
Assessment Weighting:	100%
Issue Date:	18th February 2022
Hand-in Date:	18th March 2022
Planned Feedback Date:	25th April 2022
Mode of Submission:	Online
Mode of Marking:	Online
Mode of Feedback:	Online
Number of copies to be submitted:	1 file should be submitted. A zip file containing the software artefact.

Problem Scenario

[Please note that Solent Excursions is fictional and invented for this case study.]

Solent Excursions is an organisation with their headquarters based in Southampton and with branches in cities throughout the UK. Solent Excursions specialises in arranging day excursions for tourists to local points of interests and nearby heritage sites. They help their customers by arranging excursion packages that include selecting a route and one of several pick-up points along the selected route as well as seats on the excursion vehicle (a coach, minibus, or private car).

Solent Excursions operates from its offices. Customers visit the offices in person and talk to advisors. Customers discuss their preferences with a Solent Excursions advisor who makes bookings on behalf of the customer. At present, this process is manual and involves the advisor making phone calls to check what routes and pick points as well as excursion vehicles are available and then book as appropriate.

Solent Excursions wish to improve the process of booking day excursions packages by computerising the process. They would like a software application that can be installed at each of their sites, and which allows the advisor to view available routes, pick-up points and excursion vehicles and make bookings via the application. The improved process will involve the following:

- A customer would visit one of the offices and meet with an advisor. The customer would first inform the advisor of the route they are interested in. The customer will then select a pick-up point on the route. The customer would also inform the advisor of the type of excursion vehicle they require. This can be a coach, mini-bus, or a private car. The advisor would check if the desired route, pick-up point, and excursion vehicle is available. In the case of a coach or minibus, this involves checking the required seats are available. In the case of a private car, this involves checking that a suitable sized vehicle is available. For all vehicles, the advisor should also be able to check if the vehicle has disabled access. The advisor would book the package for the customer if a suitable package is available.

- Once a package has been booked, the advisor would provide the customer with a printed summary of their package.

- An administrator should be able to add/remove routes, pick-up points, and excursion vehicles to/from the system. The admin should also be able edit these.

You have been tasked with creating the software application. The application should do the following:

- The application should be a standalone, desktop application. No web functionality or database is required.

- The application should demonstrate the application of object-oriented design and development concepts. You should follow the ICONIX process.

- The application should have a graphical user interface.

- The application should allow a customer to view their booked package using the graphical user interface.

- The application should allow an advisor to search for routes, pick-up points, and excursion vehicles, check their availability and book them if they are available through the graphical user interface.

- The application should allow an advisor to display the summary of a package using the graphical user interface. The summary will include information related to the point of interest or heritage site, the route to be taken, the pick-up point and the excursion vehicle that will be used.

- The application should allow an administrator to add/edit/remove routes, pick-up points, and excursion vehicles using the graphical user interface.

- The application should allow the data stored by the system to be exported to a single file using a suitable file format.

- The application should be tested suitably to ensure that it functions as expected.

Environment

You are required to use the following tools:
- PyCharm as your integrated development environment
- Python 3.9 as the standard python library
- Additionally, the following libraries/modules may be imported and utilised:

    csv – to process CSV files
    datatime – to manipulate dates and times
    enum – to add enumerations
    json – to process JSON files
    math – provides access to mathematical functions
    os – to retrieve or check file paths
    pprint – for 'pretty printing' data structures
    random – to generate random numbers
    test – allows regression testing
    tkinter – to create a graphical user interface
    typing – to add type hints
    unittest – to construct and run tests

- Git Tools and GitHub for version control

- A diagramming tool such as draw.io - No other python libraries or modules should be used.

Environment

Before You Begin:

You should read this assessment brief in its entirety. Once you are ready, create a new Python project using PyCharm. You should ensure that the interpreter is set to Python 3.8+. You should also ensure that your project is added to a **private** Git repository hosted on GitHub. **Failure to do so will result in you failing the assessment.**

Software Product:

You are required to develop a software product that addresses the problem scenario using Python and the tools specified in this assessment brief. The solution must demonstrate the application of object-oriented design and development (as taught in the module). You should regularly commit your project to your Git repository. It is expected that your Git repository will evidence the full development history of your software product.

Demonstration:

You will be required to demonstrate your final working solution. During the demonstration, it is expected that you will execute your final solution, explain your implementation, answer any questions related to your solution and show evidence of the development of the software artefact. No presentations slides are expected or required. The demonstrations will take place after the submission of the software product.

Assessment Criteria

Your assessment will be graded according to the following criteria:

Analysis and Design (30%)	**Grade D: Work mostly complete; significant inaccuracies and/or inconsistencies in your analysis and design. Up to 3 use-cases presented.** **Demonstration: Basic explanations of design are provided.** **Grade C: In addition to meeting the requirements for Grade D, the work is complete, diagrams predominantly correct and consistent with each other, but with some inaccuracies. 5 or more use-cases are presented. Version controlled repository shows some evidence analysis and design.** **Demonstration: Suitably demonstrates the application of suitable analysis and design techniques.** **Grade B: In addition to meeting the requirements for Grade C, the work is complete, analysis and design correct and artefacts all consistent with each other. Most use cases are presented. Version controlled repository shows evidence of incremental analysis and design.** **Demonstration: Strong explanation and understanding of analysis and design.** **Grade A: In addition to meeting the requirements for Grade B, the work is fully completed, artefacts are all consistent with each other. All use cases have been addressed. Version controlled repository shows comprehensive evidence of analysis and design.** **Demonstration: Comprehensive explanation and understanding of analysis and design.**
Implementation (40%)	**Grade D: A few use cases have been implemented. Implementation matches the design.** **Demonstration: Implementation is suitably explained.** **Grade C: In addition to the requirements for Grade D, several use cases implemented. Little error handling. Little evidence of the use of more advanced implementation technologies covered in the module. Implementation matches the design. Version controlled repository shows evidence of incremental development.** **Demonstration: Clear explanations and suitable technical understanding.** **Grade B: In addition to meeting the requirements for Grade C, an implementation that addresses all the use cases.**

	There may be room for improvement in your implementation and error handling. Some evidence of use of the more advanced techniques covered in the module. Implementation matches the design. Version controlled repository shows evidence of incremental development throughout the life of the project. **Demonstration: Detailed answers which show strong technical understanding.** **<u>Grade A</u>: In addition to meeting the requirements for Grade B, addresses all the requirements for the software product as detailed in this assessment brief. Demonstrates application of the more advanced taught concepts. Robust error handling and a user-friendly interface. Demonstrates additional learning beyond the taught concepts of this module. Version controlled repository evidences a comprehensive development process and design artefacts are all consistent with each other.** **Demonstration: Detailed answers which show strong technical understanding and ability to critically evaluation final solution.**
Testing (30%)	**<u>Grade D:</u> A small number of unit tests undertaken with significant omissions.** **Demonstration: Describes a relevant testing strategy** **<u>Grade C</u>: In addition to the requirements for Grade D, a significant number of unit tests undertaken but with several omissions. Version controlled repository shows some evidence of testing throughout project development.** **Demonstration: Suitably explains the selection of appropriate unit tests** **<u>Grade B</u>: A wide range of unit tests undertaken. There may be a small number of omissions. Version controlled repository shows significant evidence of testing throughout project development.** **Demonstration: Suitably applies a strong testing strategy.** **<u>Grade A:</u> A comprehensive range of unit tests. Version controlled repository shows comprehensive testing throughout project development.** **Demonstration: Suitably applies a comprehensive testing strategy.**

Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the Module descriptors.

Late Submissions

Students are reminded that:

i. If this assessment is submitted late i.e. within 5 working days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;
ii. If this assessment is submitted later than 5 working days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;
iii. If this assessment is being submitted as a referred piece of work then it must be submitted by the deadline date; any Refer assessment submitted late will be regarded as a non-submission and will be awarded a zero.

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-assessment-principles-and-regulations.pdf

Extenuating Circumstances

The University's Extenuating Circumstances procedure is in place if there are genuine circumstances that may prevent a student submitting an assessment. If students are not 'fit to study', they can either request an extension to the submission deadline of 5 working days or they can request to submit the assessment at the next opportunity (Defer). In both instances students must submit an EC application with relevant evidence. If accepted by the EC Panel there will be no academic penalty for late submission or non-submission dependent on what is requested. Students are reminded that EC covers only short term issues (20 working days) and that if they experience longer term matters that impact on learning then they must contact the Student Hub for advice.

Please find a link to the EC policy below:

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2p-extenuating-circumstances.pdf

Academic Misconduct

Any submission must be students' own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University's Academic Handbook includes the definitions of all practices that will be deemed to constitute academic misconduct. Students should check this link before submitting their work.

Procedures relating to student academic misconduct are given below:

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/4l-student-academic-misconduct-procedure.pdf

Ethics Policy

The work being carried out by students must be in compliance with the Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then students will need an ethics release or an ethical approval prior to the start of the project.

The Ethics Policy is contained within Section 2S of the Academic Handbook:
https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2s-solent-university-ethics-policy.pdf

Grade marking
The University uses a letter grade scale for the marking of assessments. Unless students have been specifically informed otherwise their marked assignment will be awarded a letter grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-annex-3-assessment-regulations-grade-marking-scale.docx

Guidance for online submission through Solent Online Learning (SOL)

http://learn.solent.ac.uk/onlinesubmission